



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,573	11/02/2001	Charles S. Fenton	103036.00014	2730
67942	7590	09/06/2007	EXAMINER	
RAMAN N. DEWAN JACKSON WALKER, L.L.P. 100 CONGRESS AVENUE SUITE 1100 AUSTIN, TX 78701			POLTORAK, PIOTR	
			ART UNIT	PAPER NUMBER
			2134	
			MAIL DATE	DELIVERY MODE
			09/06/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

---

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/040,573  
Filing Date: November 02, 2001  
Appellant(s): FENTON ET AL.

**MAILED**

SEP 06 2007

**Technology Center 2100**

---

Jackson Walker  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 6/01/07, appealing from the Office action  
mailed on 8/23/06.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest as STERLING  
COMMERCE, INC. is contained in the brief.

**(2) Related Appeals and Interferences**

The brief indicated no related appeals and interferences, which directly affect  
or be directly affected by or have a bearing on the decision in the pending  
appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

No amendment after final has been filed.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal  
is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is  
correct.

**(8) Evidence Relied Upon**

- ❖ Dan et al. (U.S. Patent No. 6148290),
- ❖ Epsteine et al. (U.S. Patent No. 6684329),

- ❖ Reed et al. (U.S. Patent 6266704),
- ❖ Ashdown et al. (U.S. Patent No. 6308276),
- ❖ Dan et al. (U.S. Pub. 20020178103),
- ❖ Dan et al. (U.S. Pub. 20020178103),
- ❖ Pfleeger (Charles P. Pfleeger, "Security in computing", 2nd edition, 1996, ISBN: 0133374866).

### **(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

#### ***Claim Rejections - 35 USC § 103***

Claims 1-4, 14, 24-26, 28-29, 37, 41, 43, 52-54 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290 in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329.

As per claims 1 '290 teach generating a plurality of virtual private proxies based on an agreement between a first entity and a second entity and associating a first virtual private proxy associated with the first entity and a second virtual private proxy associated with the second entity ('290, col. 5. lines 49-63 and col. 6 lines 11-25).

'290 teach monitoring data at received at the first virtual private proxy from the first entity, determining whether the data violates the agreement ('290, col. 6 lines 25-47).

'290 do not explicitly teach disallowing communication of the data from the first virtual private proxy to the second virtual private proxy when proxy when data violation is detected.

'329 teach that data is monitored to determine any violation and disallows communication of the data from the first virtual private proxy to the second virtual private proxy when proxy when data violation is detected ('329, *col. 8 line 56- col. 9 line 23*).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to disallow communication between proxies when the data violation is detected as taught by '329. One of ordinary skill in the art would have been motivated to perform such a modification in order to allow only traffic conforming to a predetermined security policy.

Claims 14, 26 and 41 are substantially equivalent to claim 1; therefore claims 14, 26 and 41 are similarly rejected.

As per claims 29 the examiner considers the second entity as a secure switch; thus the first virtual private proxy comprises a logical representation of a logical access point between the first entity and a secure switch. In order to activate the logical access point the logical access point must be accessed and software accesses entities such as access point using a logical representation of the entity;

thus the first virtual private proxy must comprise a logical representation of a logical access point. Also, the first virtual private proxy that comprises a logical representation of a logical access point is connected with the secure switch and through physical means such as communication line 532 that in networks discussed by '290 (*Background of the invention*) are implemented by physical lines. Another words, the logical representation of the logical access point between the first virtual private proxy and the secure switch is implemented by a physical access (*means*) point between the first entity and the secure switch.

As per claims 2-4, 24-25, 52-53, '290 and '329 do not explicitly teach determining that the data includes a security violation such as a virus, malicious program or an intrusion attempt and prohibiting this type of data.

Official Notice is taken that it is old and well-known practice to determine whether the data includes a security violation such as a virus, malicious program or an intrusion attempt and prohibiting this type of data. It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to determine whether the data includes a security violation such as a virus, malicious program or an intrusion attempt and prohibiting this type of data. One of ordinary skill in the art would have been motivated to perform such a modification in order to prevent security problems such as attack, loss of data etc.

Art Unit: 2134

Although in the rejection above the examiner considered that the second entity comprising the second virtual private proxies reads on a secure switch, employing an independent third party that ensures non-bias security transactions is old and well known in the computer arts. Thus, implementing the first and the second virtual proxy on an additional secure switch rather than on the first and second entity would be an obvious modification of '290 invention given a benefit of non-bias execution of agreement rules by an independent party (a secure switch).

Claims 5 and 47 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290 in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329 and further in view of Reed et al. (U.S. Patent 6266704).

'290 in view of '329 disclose the first and the second virtual private proxy as discussed above.

'290 in view of '329 do not teach hiding the existence of objects, in particular at least one of the first virtual private proxy or the second virtual private proxy.

Reed et al. teaches hiding objects and it would have been obvious to one of ordinary skill in the art at the time of applicant's invention to hide objects such as the first virtual private proxy and the second virtual private proxy. One of ordinary skill in the art would have been motivated to hide objects such as the first virtual

Art Unit: 2134

private proxy and the second virtual private proxy in order to prevent eavesdropping (*Reed et al., Abstract*).

Claim 5 and 47 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290 in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329 and further in view of Pfleeger (Charles P. Pfleeger, "Security in computing", 2nd edition, 1996, ISBN: 0133374866).

'290 in view of '329 disclose the first and the second virtual private proxy as discussed above.

'290 in view of '329 do not teach hiding the existence of objects, in particular at least one of the first virtual private proxy or the second virtual private proxy.

Pfleeger teaches hiding objects (need-to-know rule, e.g. pg. 271) and it would have been obvious to one of ordinary skill in the art at the time of applicant's invention to hide objects such as the first virtual private proxy and the second virtual private proxy given the benefit of increased security.

Claims 38-39 and 53 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290, in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329 and further in view of Ashdown et al. (U.S. Patent No. 6308276), hereinafter '276.



Art Unit: 2134

'290 teach logging violations ('290, *col. 6 lines 48-56*) and '329 teach alarms and reporting that is associated with data filtering ('329, *col. 10 lines 32-65*).

As per claims 38-39 and 53 '290 and '329 do not explicitly teach generating an alarm based on the violation, 5 discarding the data that violates the agreement 3 and communicating the alarm to a system administrator.

'276 teach (in addition to logging the violation) discarding the data that violates the agreement and alarms reported to a system administrator ('276, *col. 1 lines 29-45, col. 3 lines 1-6, Fig. 7, col. 9 lines 12-42, col. 11 lines 63-67*).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to implement logging the violation, discarding the data that violates the agreement and alarms reported to a system administrator as taught by Ashdown et al. One of ordinary skill in the art would have been motivated to perform such a modification in order to completely control the data flow.

Claims 8-10, 21-23, 27, 33-36, 42, and 55 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290 in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329 and further in view of Dan et al. (U.S. Pub. 20020178103) hereinafter '103.

'290 and '329 teach data exchange between entities utilizing the virtual private proxies, wherein data is filtered based on the agreement as discussed above.

As per claim 8, 10, 21, 23, 27, 33-36, 42, 55, '290 and '329 do not explicitly teach that the entity comprise business, do not teach generating the agreement based on two profiles that are associated with the communicating entities and that are used to generate the agreement, and do not teach that profiles comprise name and contact information, a transport protocol and messaging protocol and process specification document.[32 and 35].

'103 teach two business entities [1] with profiles comprising name and contact information generating an agreement based on two profiles associated with the communicating entities [38], the profiles comprising name and contact information [35] and messaging protocol [33].

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to use business profiles to generate an agreement as taught by '103. One of ordinary skill in the art would have been motivated to perform such a modification in order to easily negotiate a contract based on the advertised businesses capability.

As per claims 9, 22, '290, '329 and '103 do not teach that the profiles comprise a transport security protocol and that the data allowed comprise XML data.

Official Notice is taken that transport security protocols (*e.g. IPSec, PPTP, LT2P etc.*) as well as XML data are and well-known and utilized in data communication between entities. Utilizing these protocols are obvious variations that are well known in the art. One would have been motivated to include these protocols in

profiles and include XML data in allowed data especially in light of the benefits of these protocols and data as evidenced by their commercial success.

Claims 48-51 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290 in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329 and Pfleeger (Charles P. Pfleeger, "Security in computing", 2nd edition, 1996, ISBN: 0133374866) and further in view of Dan et al. (U.S. Pub. 20020178103) hereinafter '103.

Claims 48-51 introduce substantially equivalent limitations to limitations of claims 33-36; therefore claims 48-51 are similarly rejected.

Claims 48-51 are rejected under 35 U.S.C. 103(a) unpatentable over Dan et al. (U.S. Patent No. 6148290), hereinafter '290 in view of Epsteine et al. (U.S. Patent No. 6684329), hereinafter '329 and Reed et al. (U.S. Patent 6266704) and further in view of Dan et al. (U.S. Pub. 20020178103) hereinafter '103.

Claims 48-51 introduce substantially equivalent limitations to limitations of claims 33-36; therefore claims 48-51 are similarly rejected.

#### **(10) Response to Argument**

Art Unit: 2134

*On pages 5-10 appellant contests independent claims 1, 14, 26 and 41. Appellant argues that Dan does not teach private proxies. It appears that the main argument is presented in the last paragraph of pg. 5, wherein appellant states: "whereas it is axiomatic that the claimed VPPs are private, Dan's enforcement code components are components of a business service application that Dan explicitly describes and praises as being public".*

However, appellant arguments, and in particular conclusions drawn from loosely presented arguments, are not persuasive.

Dan's invention includes a plurality of virtual proxies (e.g. 502, 504 and 506) for enforcing particular service contracts and services on behalf of private entities ("the clients and participating business services may all be owned by different organizations with different degrees of understanding and trust of each other", col. 5 lines 42-48 and lines 53-55) that result in "the owner and the provider of business services" controlling "service implementation component that executes ... entirely on the service execution engine of the business service provider" (col. 5 line 49-col. 6 line 10).

Furthermore, objects 416 and 430 in Fig. 4 and associated text (col. 5 lines 29-32, Fig. 4 etc.) disclose that the invention is directed towards "a public access or enterprise network". Thus, it is clear that Dan's invention can equally be

Art Unit: 2134

implemented in public as well as private environment, and the name itself (private/public) would not affect the functionality of the invention.

Additionally, the examiner points out that the specification does not provide a definition of the term "private" in regard to the term "virtual private proxies". All resources, in particular resources that need maintenance/development/administration have "an owner". Dan discloses that virtual proxy services are implemented on a server, for example (col. 5 line 26), and an ordinary artisan would readily recognize that administration and maintenance of servers, updates and development of codes implemented on the server takes time and costs money.

*Appellant argues that elements of Dan's system and network are not proxies because "...in the field of computer networks, a proxy machine or proxy server is a computer that offers a computer network service to allow client to make indirect network connections to other network services" and that "one of ordinary skill would readily recognize a distinction between a software applicant ... and a proxy as claimed."*

Once again, appellant attempts to provide an interpretation that would aid appellant's assertion. However, the specification does not provide a definition of a virtual proxy, and appellant attempts to present a "proxy server" as a definition of a "proxy". In particular, appellant provides an example found in Wikipedia ("a client connects to the

Art Unit: 2134

proxy server, then requests a connection, file, or other resource available on a different server. The proxy provides the resource either by connecting to the specified server or by serving it from a cache"). This definition presented by appellant, refers to a particular (not virtual) service of a network device.

The examiner points out that the claim language does not limit proxies to "computers", Furthermore, one of ordinary skill in the art of computing would readily recognize that such a restriction (restricting entities providing particular functionalities to hardware) disregards the reality of computing systems. For example, Microsoft Proxy Server, first offered by Microsoft Corporation in January 1997 for Windows NT 4.0 platform, is an example of a software product providing proxy services. Also, see Epstein's disclosure in col. 1 lines 43-57, that recites: "Application-level firewall proxies are fragile, and are growing ever more complex ... As the number of protocols increases, proxies are increasingly written by people without sufficient training in writing safe software...").

The examiner also points out that the claim limitations require not private proxies but "virtual private proxies". Consulting Webster's "The computer and Internet Dictionary", 3<sup>rd</sup> edition (ISBN: 0375703519, not included) the examiner verified definition of "virtual" to be defined as "Not real. The term *virtual* is popular among computer scientists and is used in a wide variety of situations. In general, it distinguishes something that is merely conceptual from something that has physical reality".

Art Unit: 2134

*On pg. 6, appellant continues that "there is no motivation to modify Dan's public system, and that to implement Epstein's system "would undermine a basic operating principle of Dan... that is ... intended to be implemented across publicly available networks".*

Appellant does not offer any arguments to validate the assertion that "there is no motivation to modify" Dan's invention according to Epstein's disclosure and that "implementing Epstein's system would undermine a basic operating principle of Dan's invention". As a result, the examiner is not sure what arguments would be sufficient to address appellant's assertion.

The examiner points out that the motivation to combine Dan's invention with Epstein's disclosure has been offered in the final Office Action. However, in case that the format provided in the Final Office Action confused or mislead appellant in identifying relevant reading in Dan's and Epstein's invention (in particular to compatibility of these two disclosures), the examiner provides alternative step by step mappings of these two disclosures in the Conclusion, below.

*Appellant suggests (pg. 6-8) that Dan in view of Epsteine fails to disclose "determining whether the data violates the agreement" and "disallowing communication of the data from the first virtual private proxy to the second virtual private proxy when the data violates the agreement".*

The examiner points out that in addition to Dan disclosing using an agreement and taking appropriate action based on the agreement data to be communicated from first entity utilizing a first proxy and second entity utilizing a second proxy (Dan, col. 6 lines 11-24), Epsteine suggests allowing or disallowing data communication based on a predetermined agreement (security policy, col. 1 lines 15-26).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to disallow communication of the data between the first entity (using a first virtual proxy) and the second entity (using the second virtual proxy) disclosed by Dan, based on an agreement as disclosed by Epsteine. One of ordinary skill in the art would have been motivated to perform such a modification in order to allow only traffic conforming to a predetermined security policy.

See, the Conclusion for details.

*On page 10 appellant contests claim 8 rejected as unpatentable over Dan in view of Epsteine, and further in view of Dan. Specifically, appellant argues that "the Office Action does not allege that Dan '103 teaches an agreement that includes the types of data allowed".*

Appellant's arguments, carefully considered, are found not persuasive. Dan in view of Epsteine compares data against parameters found in an agreement (e.g. "determining whether the data violates the agreement"). Thus, it is implicit that the types of data that is



Art Unit: 2134

allowed must be listed. Additionally, Dan '103, provides exemplary types of data that could be found in an agreement (e.g. Fig. 3 and associated text in paragraph 32). The examiner points out that the type of protocols used (which inherently identifies data type format that could be used in communication) as disclosed explicitly by Dan '103 (paragraph 3: SMTP, HTTP, etc. are example of protocols types corresponding to object 150 Transport Protocol in Fig. 3 that could be used to communicate with a particular entity) could be used to define further details to filter data based on data type, such as protocols types (see e.g. Epsteine, col. 1 lines 15-43).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to include allowable data types in an agreement as disclosed by Epsteine given the benefit of fine control of data communication.

*On page 11 appellant contests rejections of claim 55. However, appellant does not offer any additional arguments beyond the previously addressed allegations.*

The examiner points appellant to previously stated examiner's responses, above.

**Conclusion:**

**Clients 404-408 and business service 400 disclosed by Dan in Fig. 4, read on first and second entity.**

Art Unit: 2134

DAN discloses that a business service 400 is provided in a networked environment and is implemented as a computer program. Clients 404-408 (a first entities) make requests to this business service 400 (a second entity). The clients typically execute on workstations and PCs, which reach the business service engine 402 (e.g., a server) on which the business service program 400 is provided (col. 5 lines 13-26).

Business service application 500 corresponds to the business service 400 (col. 5 lines 49-52).

**The contract 514, disclosed in Fig. 5, for example, reads on an agreement.**

The contract 514 (an agreement) can be created and owned by the provider of a service (510) or jointly created (e.g. through negotiation) by the provider and the client using the service (col. 6 lines 11-18).

**In order to send data over network entities must utilize a set of processes enabling network connection (e.g. processes receiving a network connection request, creating communication ports, keep track of and utilizing ports for communicate data etc.). In the case of Dan's invention, the set of processes also include enforcement code components 512 and 502 (associated with the first and the second entity). These read on virtual proxies.**

Art Unit: 2134

The service contract 514 specifies all the permitted interaction patterns by the client and expresses the required interaction pattern behaviors of the service provider: it provides for a self-enforcing mechanism for managing the service transactions by providing for enforcement code (or modules) to be written by the respective parties according to the rules of interaction included in the service contract (col. 6 lines 11-24).

The enforcement code can be generated automatically from the contract and tools can be provided to automatically generate enforcement code components 512 (a first virtual proxy component) and 502 (a second virtual proxy component), which will execute in the client engine as the client contract enforcer component 512 and, in the server engine, as the server contract enforcer component 502 (col. 6 lines 26-34).

*Dan discloses generating a plurality of virtual private proxies (enforcement code components) based on an agreement (a contract service) between a first entity (a client) and a second entity (a server), associating a first virtual private proxy of the plurality of virtual private proxies with the first entity and a second virtual private proxy of the plurality of virtual private proxies with the second entity.*

---

The applications 526 and 500 interact with each other via communication line 532 (col. 6 lines 45-47). The client/requester logic implementation 528 executing in the client engine 516, makes its service requests via an interface 530, which is a standard programming interface identifying the types of requests for service which can be made for the service provided by the application 500. This interface 530

Art Unit: 2134

actually passes the requests to the generated client enforcement code component 512.

According to the present invention, the enforcement code components can serve many purposes in the function of enforcing the specifications of the service contract.

For example, enforcement code 512 upon receiving a request to be sent from the application 526 can log the request ... and pass the request by a chosen protocol.

When receiving a request or response from the service application 500, the enforcement code component can provide some of the functions listed hereinabove and also can determine whether the message is a response or a request, check validity of response and take appropriate action (col. 6 lines 62-67) according to the rules of interaction included in the service contract (col. 6 lines 11-24).

*Dan's virtual private proxies are private proxies.*

---

The service contract (the agreement) of the invention has most immediate value in the context of providing a business service on a public network but it can be applied to other environments (col. 5 lines 13-16).

The clients and participating business services may all be owned by different organizations with different degrees of understanding and trust of each other (col. 5 lines 42-48 and lines 53-55).

Art Unit: 2134

DAN does not disclose disallowing communication of data from the first virtual private proxy to the second virtual private proxy when proxy data violation is detected.

Epsteine discloses firewalls a firewall system 120 operative to screen all connections between private network 110 and untrusted system 140. These connections are facilitated by Internet network 130. In the screening process, firewall system 120 determines which traffic should be allowed and which traffic should be disallowed based on a predetermined security policy (Fig. 1, and col. 1 lines 15-26).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to disallow communication of the data between the first entity (using a first virtual proxy) and the second entity (using the second virtual proxy) disclosed by Dan, based on an agreement as disclosed by Epsteine. One of ordinary skill in the art would have been motivated to perform such a modification in order to allow only traffic conforming to a predetermined security policy.

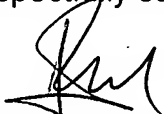
**Thus, Dan in View of Epstein clearly discloses the limitations of argued claims 1, 14, 26 and 41.**

#### **(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,



Peter Poltorak

Conferees:



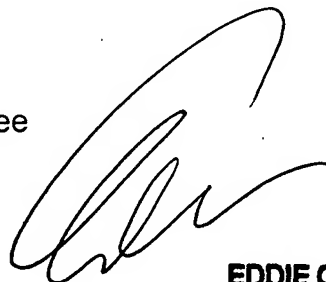
KAMBIZ ZAND  
SUPERVISORY PATENT EXAMINER

Kambiz Zand



KAMBIZ ZAND  
SUPERVISORY PATENT EXAMINER

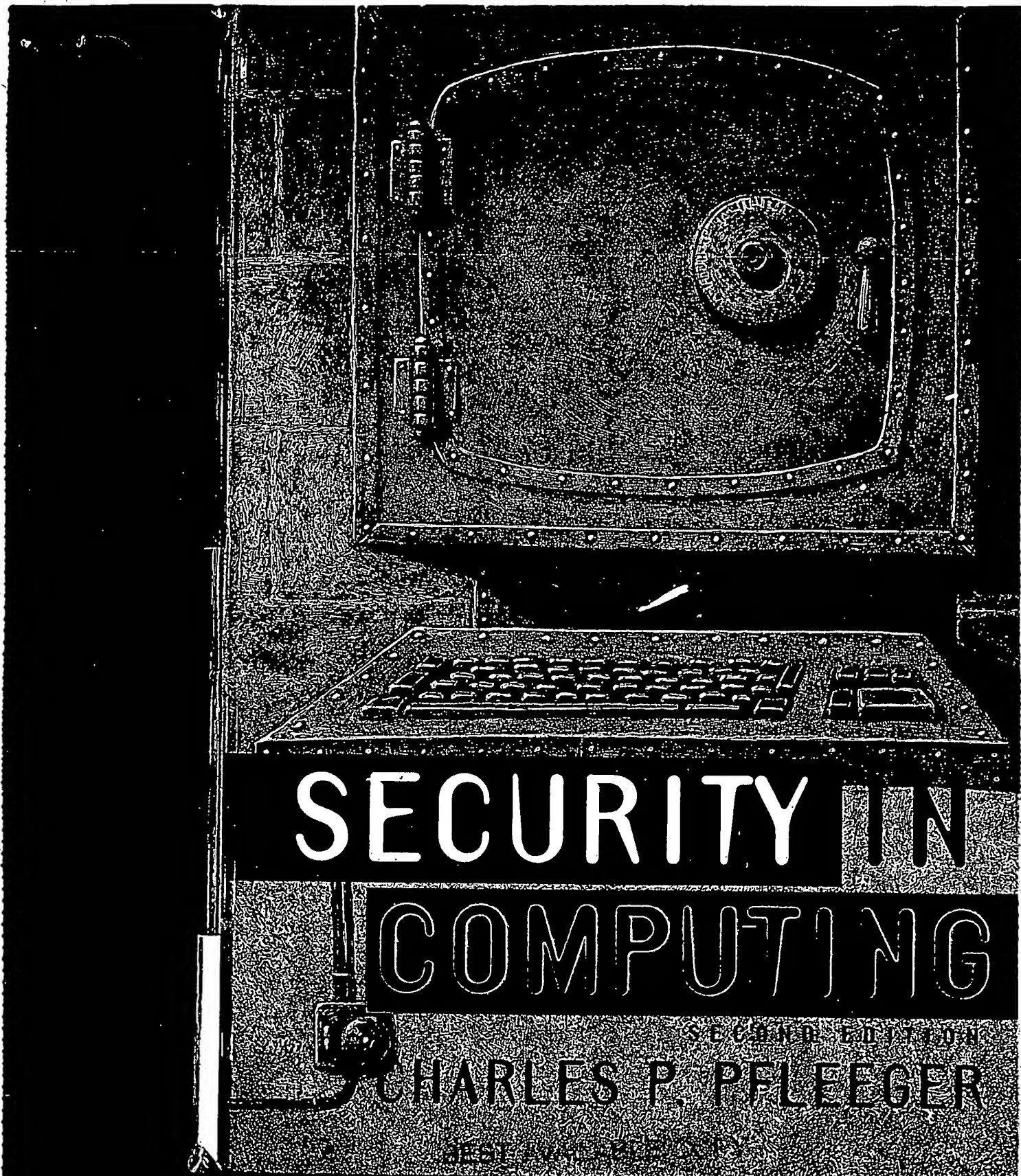
Eddie Lee



EDDIE C. LEE  
SUPERVISORY PATENT EXAMINER



UNITED STATES PATENT AND TRADEMARK OFFICE



# SECURITY IN COMPUTING

SECOND EDITION  
CHARLES P. PFLIEGER

WILEY-INTERSCIENCE

Library of Congress Cataloging-in-Publication Data

Pfleeger, Charles P., 1948-

Security in computing / Charles P. Pfleeger. — Rev. ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-13-337486-6

1. Computer security. 2. Data protection. 3. Privacy, Right of.

I. Title.

QA76.9.A25P45 1996

005.8—dc20

96-32910

CIP

Editorial Production: *Precision Graphic Services, Inc.*

Acquisitions Editor: *Paul W. Becker*

Manufacturing Manager: *Alexis R. Heydt*

Cover Design Director: *Jerry Votta*

Cover Design: *Moran Design, Inc.*



© 1997 by Prentice Hall PTR

Prentice-Hall, Inc.

A Simon & Schuster Company

Upper Saddle River, NJ 07458

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact:

Corporate Sales Department

Prentice Hall PTR

1 Lake Street

Upper Saddle River, NJ 07458

Phone: 800-382-3419

FAX: 201-236-7141

E-mail: [corpsales@prenhall.com](mailto:corpsales@prenhall.com)

Printed in the United States of America

10 9 8 7 6 5 4

ISBN: 0-13-337486-6

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada, Inc., *Toronto*

Prentice-Hall Hispanoamericana S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

S

Table 3-6

Box	Row
S <sub>1</sub>	0
	1
	2
	3
S <sub>2</sub>	0
	1
	2
	3
S <sub>3</sub>	0
	1
	2
	3
S <sub>4</sub>	0
	1
	2
	3
S <sub>5</sub>	0
	1
	2
	3
S <sub>6</sub>	0
	1
	2
	3
S <sub>7</sub>	0
	1
	2
	3
S <sub>8</sub>	0
	1
	2
	3

BEST AVAILABLE COPY



- *Model.* Before beginning to create a trusted operating system, the designer must have confidence that the proposed system will meet its requirements. The designer constructs a model of the environment to be secured and studies different ways of enforcing that security. In the second part of this chapter we consider several different models for operating system security. The model is actually a representation of the policy that the operating system will enforce.
- *Design.* After having selected a model of security, the designer must choose a means to implement that model. The design covers both what the trusted operating system is and how it is to be constructed. The third major section of this chapter addresses choices that can be made during development of a trusted operating system.
- *Trust.* Because the operating system bears such a central role in enforcing security, we want some basis for believing that it will meet our expectations. Trust is based on two aspects: *features* (the operating system has all the necessary functionality needed to enforce the expected security policy) and *assurance* (the operating system has been implemented in such a way that we have confidence it will enforce the security policy). In the fourth part of this chapter we explore what makes a particular design or implementation worthy of trust.

Finally, we look at some examples. Several trusted operating systems have been written, and more are under development. Some secure systems were originally designed for security; in others, security features were added to existing operating systems. The fifth part of this chapter presents examples of both of these ways to produce a secure operating system.

## 7.1 What Is a Trusted System?

Before we begin to examine these parts in detail, however, we should develop a common understanding of important terms. What would it take to consider something secure? The word *secure* is binary: something either is or is not secure. If it is secure, it should withstand all attacks, today, tomorrow, and a century from now. And if I sell you a secure lock, for example, it is my assertion that it is secure; you either accept my assertion (and buy the lock) or reject it (and not buy). Let's move from *secure* to another, similar adjective: *good*. If I am selling a "good" used car, you are less interested in my thinking it good ("I really did enjoy driving that car; it was a good one") than you are in a fair appraisal of its condition and being able to judge for yourself that it meets your needs.

For reasons such as the ones in the previous paragraph, security professionals speak of *trusted* rather than *secure* operating systems, connoting ones that meet their intended security requirements, are of high enough quality, and justify confidence in their quality. Trust is a quality of the receiver, not of the giver. I would not say my used car is trusted; you trust my description, your evaluation, or the opinion of a friend or a mechanic. But in the end, the responsibility is yours to develop the level of trust you require.

Also there can be degrees of trust: you trust certain friends with deep secrets, but others you trust only to give you the correct time of day. You develop trust based on evidence and experience: banks increase their trust in borrowers as the borrowers repay loans as expected, so that borrowers with good trust (credit) records can borrow larger amounts. Finally, trust is earned, not claimed or conferred. The comparison in Table 7-1 highlights some of these distinctions.

Tr  
—  
Se  
—  
• /  
• /  
• /  
• /  
• /  
• /  
—

Yi

process  
capable  
product  
security  
ing sys  
security  
cient h  
informa  
tions yo

- en
- su
- ev

In  
them tr

## 7.2 Secur

In order  
able to s  
to enfor  
in relativ  
We  
of the w  
precisely

### Militar

The mili  
informat  
dential,  
rest of th  
ing order  
Infi  
allowed

er must have  
designer con-  
ays of enforc-  
ifferent mod-  
of the policy

oose a means  
ting system is  
pter addresses  
tem.

ig security, we  
s based on two  
dity needed to  
stem has been  
e security pol-  
ular design or

ave been writ-  
y designed for  
. The fifth part  
erating system.

elop a common  
ng secure? The  
it should with-  
u a secure lock,  
on (and buy the  
adjective: *good*.  
good ("I really  
sal of its condi-

essionals speak  
at their intended  
in their quality.  
ed car is trusted;  
mechanic. But in

secrets, but oth-  
sed on evidence  
s repay loans as  
larger amounts.  
le 7-1 highlights

Table 7-1 Qualities of Security and Trustedness

Secure	Trusted
<ul style="list-style-type: none"> <li>• <i>Either-or</i>: Something either is or is not secure</li> <li>• <i>Property of presenter</i></li> <li>• <i>Asserted</i>: based on product characteristics</li> <li>• <i>Absolute</i>: not qualified as to how, where, when, or by whom used</li> <li>• <i>A goal</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Graded</i>: There are degrees of "trustedness"</li> <li>• <i>Property of receiver</i></li> <li>• <i>Judged</i>: based on evidence and analysis</li> <li>• <i>Relative</i>: viewed in context of use</li> <li>• <i>A characteristic</i></li> </ul>

You will see the adjective *trusted* many times in this chapter, as in *trusted process* (a process that can affect system security; a process whose incorrect or malicious execution is capable of violating system security policy), *trusted product* (an evaluated and approved product), *trusted software* (the software portion of a system that can be relied on to enforce security policy), *trusted computing base* (the set of all protection mechanisms in a computing system, including hardware, firmware, and software, that together enforce a unified security policy over a product or system), or *trusted system* (a system that employs sufficient hardware and software integrity measures to allow its use for processing sensitive information). These definitions are paraphrased from [NIS91b]. Common to these definitions you can see the concepts of

- enforcement of security policy
- sufficient measures and mechanisms
- evaluation

In this chapter we study trusted operating systems, and examine closely what makes them trustworthy.

## 7.2 Security Policies

In order to know that an operating system maintains the security we expect, we have to be able to state what that security is. A **policy** is a statement of the security we expect the system to enforce. An operating system (or any other piece of a trusted system) can be trusted only in relation to a security policy, that is, to the security needs the system is expected to satisfy.

We begin by studying the military security policy because it has been the basis of much of the work in development of trusted operating systems, and because it is fairly easy to state precisely. Then we move to security policies that commercial establishments might adopt.

### Military Security Policy

The military security policy is based on protecting classified information. Each piece of information is ranked at a particular sensitivity level, such as *unclassified*, *restricted*, *confidential*, *secret*, or *top secret*. We can denote the sensitivity of an object *O* by *rank<sub>O</sub>*. In the rest of this chapter we assume these five sensitivity levels. The ranks are shown in increasing order of sensitivity, as in Figure 7-1.

Information access is limited by the **need-to-know** rule: access to sensitive data is allowed only to subjects who need to know that data to perform their jobs. Each piece of

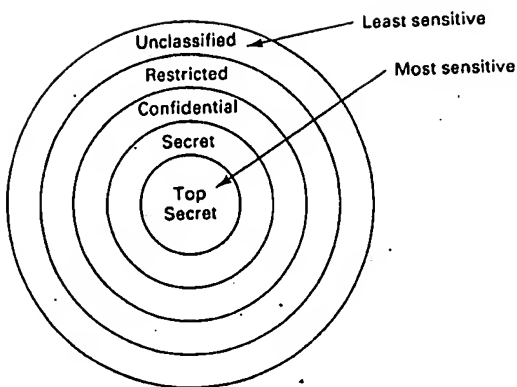


Figure 7-1 Hierarchy of Sensitivities

classified information may be associated with one or more projects, called **compartments**, describing the subject matter of the information. Compartments help to enforce need-to-know restrictions so that people can obtain access only to information that is relevant to their jobs. A compartment may cover information at only one sensitivity level, or it may include information of more than one sensitivity level. The relationship between compartments and sensitivity levels is shown in Figure 7-2.

Examples of compartment names might be *snowshoe*, *crypto*, and *Sweden*. A single piece of information is coded with zero, one, two, or more compartments, depending on the categories to which it relates. The association of information and compartments is shown in Figure 7-3. For example, one piece of information may be a list of publications on cryptography, while another may describe development of snowshoes in Sweden. The compartment of this first piece of information is {*crypto*}; the second is {*snowshoe*, *Sweden*}.

The combination  $\langle \text{rank}; \text{compartments} \rangle$  is called the class or classification of a piece of information.

A person seeking access to sensitive information must be cleared. A clearance is an indication that a person is trusted to access information up to a certain level of sensitivity, and that the person needs to know certain categories of sensitive information. The clearance of a subject is a combination  $\langle \text{rank}; \text{compartments} \rangle$ . This combination has the same form as the classification of a piece of information.

Now we introduce a relation  $\leq$ , called **dominance**, on sensitive objects and subjects. For a subject  $s$  and an object  $o$ ,

$$s \leq o \text{ if and only if} \\ \text{rank}_s \leq \text{rank}_o \text{ and} \\ \text{compartments}_s \subseteq \text{compartments}_o$$

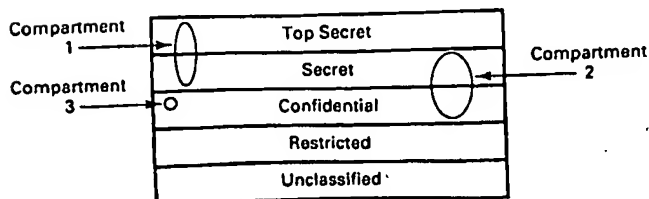
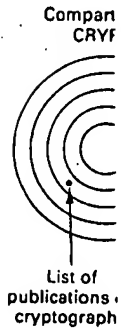


Figure 7-2 Compartments and Sensitivity Levels



We say that  
spondingly  
can access.  
least as high  
compartment  
saying that  
Inform  
to (top secret; (crypto; Sweden).  
Military  
ments. Sen.  
restrictions  
rigidly controlled  
clearances:

### Comments

The comments of the same

A large  
sions or degrees.  
There are a  
files. Data is  
internal; the  
degrees. Less  
than internal  
as people work  
projects and  
accounting  
ity. There are  
have no need  
have access  
Figure 7-4.

Two significant  
ity. First, on

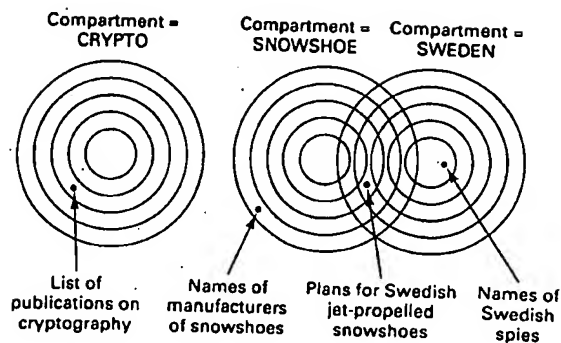


Figure 7-3 Association of Information and Compartments

We say that  $o$  dominates  $s$  (or  $s$  is dominated by  $o$ ) if  $s \leq o$ ; the relation  $\geq$  is defined correspondingly. Dominance is used to limit the sensitivity and content of information a subject can access. A subject can read an object only if (a) the clearance level of the subject is *at least as high* as that of the information, and (b) the subject has a need to know about *all* compartments for which the information is classified; these conditions are equivalent to saying that the subject dominates the object.

Information classified  $\langle \text{secret}; \{\text{Sweden}\} \rangle$  could be read by someone cleared for access to  $\langle \text{top secret}; \{\text{Sweden}\} \rangle$  or  $\langle \text{secret}; \{\text{Sweden}, \text{crypto}\} \rangle$ , but not by someone with a  $\langle \text{top secret}; \{\text{crypto}\} \rangle$  clearance or someone cleared for  $\langle \text{confidential}; \{\text{Sweden}\} \rangle$ .

Military security enforces both sensitivity requirements and need-to-know requirements. Sensitivity requirements are known as **hierarchical** requirements; need to know restrictions are **nonhierarchical**. This model is appropriate for a setting in which access is rigidly controlled by a central authority. Someone, often called a security officer, controls clearances and classifications, which are not generally up to individuals to alter.

### Commercial Security Policies

The commercial world is less rigidly and less hierarchically structured. Still, we find many of the same concepts.

A large organization, such as a corporation or a university, may be broken into divisions or departments, with each of those responsible for a number of disjoint projects. There are also some corporate-level responsibilities, such as accounting and personnel files. Data items may have different degrees of sensitivity, such as *public*, *proprietary*, or *internal*; the names vary between organizations, and so there is no universal hierarchy of degrees. Let us assume that *public* is less sensitive than *proprietary*, which is less sensitive than *internal*. Projects and departments tend to be fairly well-separated, with some overlap as people work on two or more projects. Corporate-level responsibilities tend to overlay projects and departments, as people throughout the corporation may have need for accounting or personnel data. However, even corporate data may have degrees of sensitivity. There also tends to be more sensitivity due to projects: people on project *old-standby* have no need to know about project *new-product*, although people on *new-product* may have access to all data on *old-standby*. Thus, a commercial layout of data might look like Figure 7-4.

Two significant differences exist between commercial and military information security: First, outside the military, there is no formalized notion of clearances: a person is not